# Task 1: Data Preparation and Analysis

**Task 1.1**

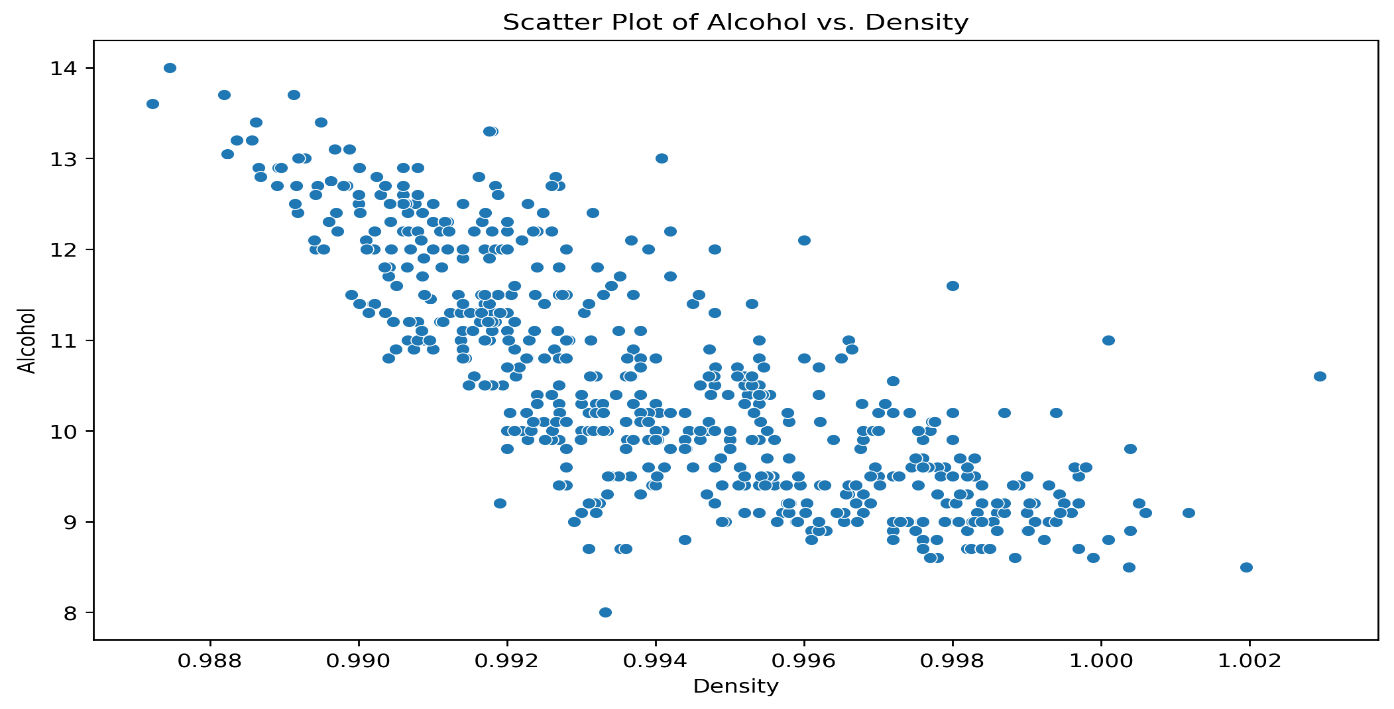
In this case, a dataset containing wine quality information is loaded and processed. Initially, the dataset is explored through functions like info(), describe(), and head() to understand its structure and contents. To address data quality issues, rows with spaces in the "free sulfur dioxide" and "total sulfur dioxide" columns are removed. Missing values are also handled by dropping rows with any NaN values.

Following data cleaning, the code verifies if a minimum of 600 rows remains in the dataset; if not, it indicates that there isn't enough data without missing values. However, if there are at least 600 rows, a random sample of 600 instances is extracted and saved as "A2RandomSample.csv." This random sample provides a representative subset for further analysis and modeling, ensuring that the dataset is suitable for subsequent tasks. The code snippet showcases essential data preprocessing steps critical for reliable and informative data analysis.

**Task 1.2: Explore the relationship between two variables: alcohol and density.**

* **Show the relationship in an appropriate graph (i.e., chart). Describe any interesting relationships (or lack of relationships) that you observe from the visualisation.**
* **Build a linear model (i.e., Simple Linear Regression) for the two variables, with alcohol being dependent variable and density as independent variable. Present the linear model in the Report and interpret the coefficients of the model.**

**Scatter Plot:** A scatter plot is created to visualize the relationship between alcohol content and density. The x-axis represents density, the y-axis represents alcohol content, and each point in the plot represents a data point from the dataset. This plot helps to visually assess the correlation between the two variables.



**Line Plot:** A line plot is generated by sorting the dataset by density. This line graph shows the trend of alcohol content concerning density, with density values on the x-axis and alcohol content on the y-axis. The line provides a visual representation of the relationship between the two variables.

A graph showing a graph

Description automatically generated with medium confidence

**Correlation Heatmap:** A correlation heatmap is created to quantify the relationships between all pairs of numerical variables in the dataset. The heatmap is annotated with correlation coefficients. In particular, it shows that the density and alcohol content exhibit a strong negative correlation of -0.80.

A screenshot of a computer

Description automatically generated

**Interpretation of Correlation:** It is explained that the negative correlation of -0.79 indicates that as the density of the substance increases, the alcohol content decreases. This interpretation is consistent across scatter plots, line graphs, and is confirmed by Pearson correlation analysis.

**Linear Regression:** A linear regression analysis is performed to model the relationship between density (independent variable) and alcohol content (dependent variable). The dataset is split into training and testing sets, and a linear regression model is fitted to the training data. Predictions are made on the test data.

**R-squared (R2) Score:** The R-squared (R2) score is calculated to evaluate the goodness of fit of the linear regression model. The R2 score is reported, indicating the proportion of the variance in alcohol content that is explained by density.

**Slope and Intercept**: The slope (coefficient) and intercept of the linear regression model are determined. The slope represents the change in alcohol content for a one-unit change in density, and the intercept is the value of alcohol content when density is zero.

The R-squared (R2) score of 0.47 implies that 47% of the variation in alcohol content can be explained by density. The negative slope of -331.45 indicates that as density increases, alcohol content tends to decrease by approximately 331.45 units. The intercept of 340.01 represents the estimated alcohol content when density is zero, although this value may not have practical significance in the context of the data.

**Task 1.3: Explore the relationship between two variables: quality and alcohol.**

* **Create the side-by-side boxplot for alcohol grouped by quality level.**
* **Summarise your findings based on the boxplot.**

The box plot reveals a positive relationship between wine quality and alcohol content. As wine quality increases, the median alcohol content tends to rise, signifying that higher-quality wines typically have higher alcohol levels. Lower-quality wines (e.g., quality ratings 3 and 4) tend to cluster with lower alcohol content, while higher-quality wines exhibit more balanced distributions. However, outliers in wine quality ratings 4, 5, and 8 indicate exceptional or unusual wines within these categories. The dataset contains counts for each unique quality rating, and the unique quality ratings are provided as well. This visualization effectively illustrates the association between wine quality and alcohol content. A screenshot of a graph

Description automatically generated

# Task 2: Classification

**Task 2.1: Select a model, either k-NN (k-Nearest Neighbours) or Decision Tree. Train and evaluate the model appropriately. Use at least 3 metrics for evaluation.**

**Decision Tree:**

**Data Preparation:**

* The dataset was divided into feature variables (X) and the target variable (y), with "quality" as the target variable.

**Hyperparameter Tuning:**

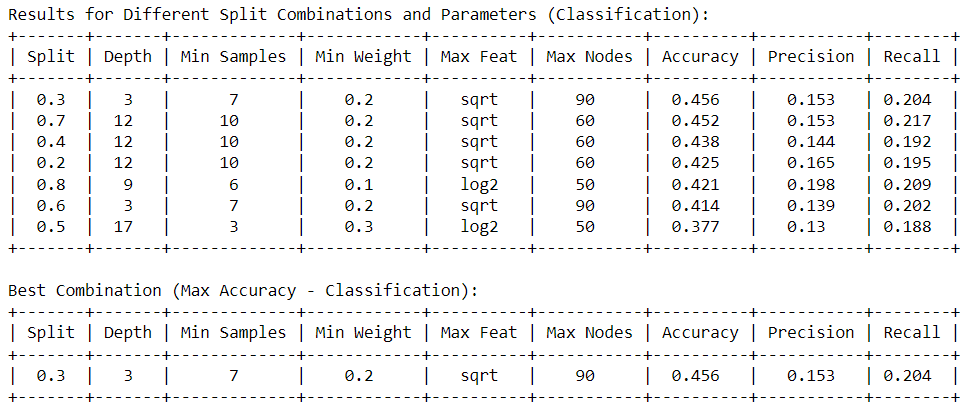
* A range of hyperparameters was considered for tuning the Decision Tree Classification model, including max\_depth, min\_samples\_leaf, min\_weight\_fraction\_leaf, max\_features, and max\_leaf\_nodes.
* A Randomized Search Cross-Validation (RandomizedSearchCV) was performed to find the best combination of hyperparameters for each training-test split. We considered different splits ranging from 20% to 80% for testing.

**Model Evaluation:**

* For each combination of hyperparameters and split, the Decision Tree model was trained and tested.
* The model's performance was evaluated using key metrics, including accuracy, precision, and recall.

**Results:**

* The best combination of hyperparameters that achieved the highest accuracy was identified for each split. Results were summarized in a tabulated format, showcasing the key hyperparameters and corresponding performance metrics for each split.



**KNN Classification:**

**Data Preprocessing:**

* The dataset was standardized using the StandardScaler, and the "quality" column was separated from the feature variables.

**Hyperparameter Tuning:**

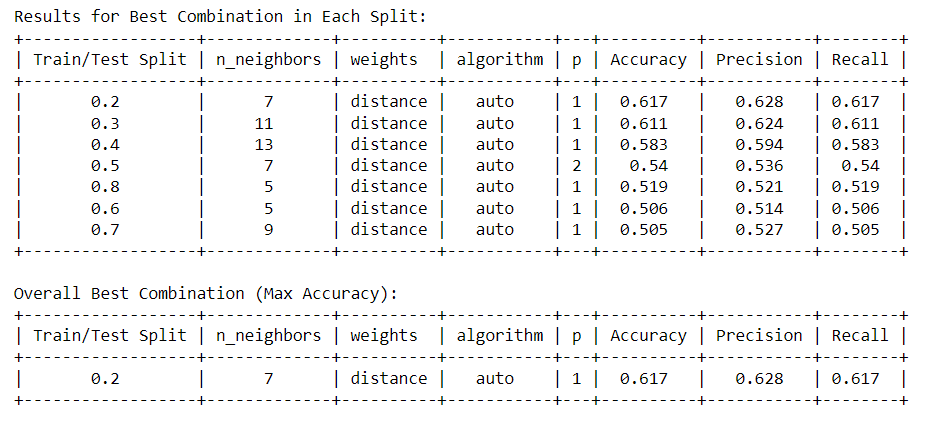
* A range of hyperparameters were considered for tuning the KNN classification model, including n\_neighbors (number of neighbors), weights (weighting of neighbors), algorithm (neighbor search algorithm), and p (distance metric: 1 for Manhattan, 2 for Euclidean).

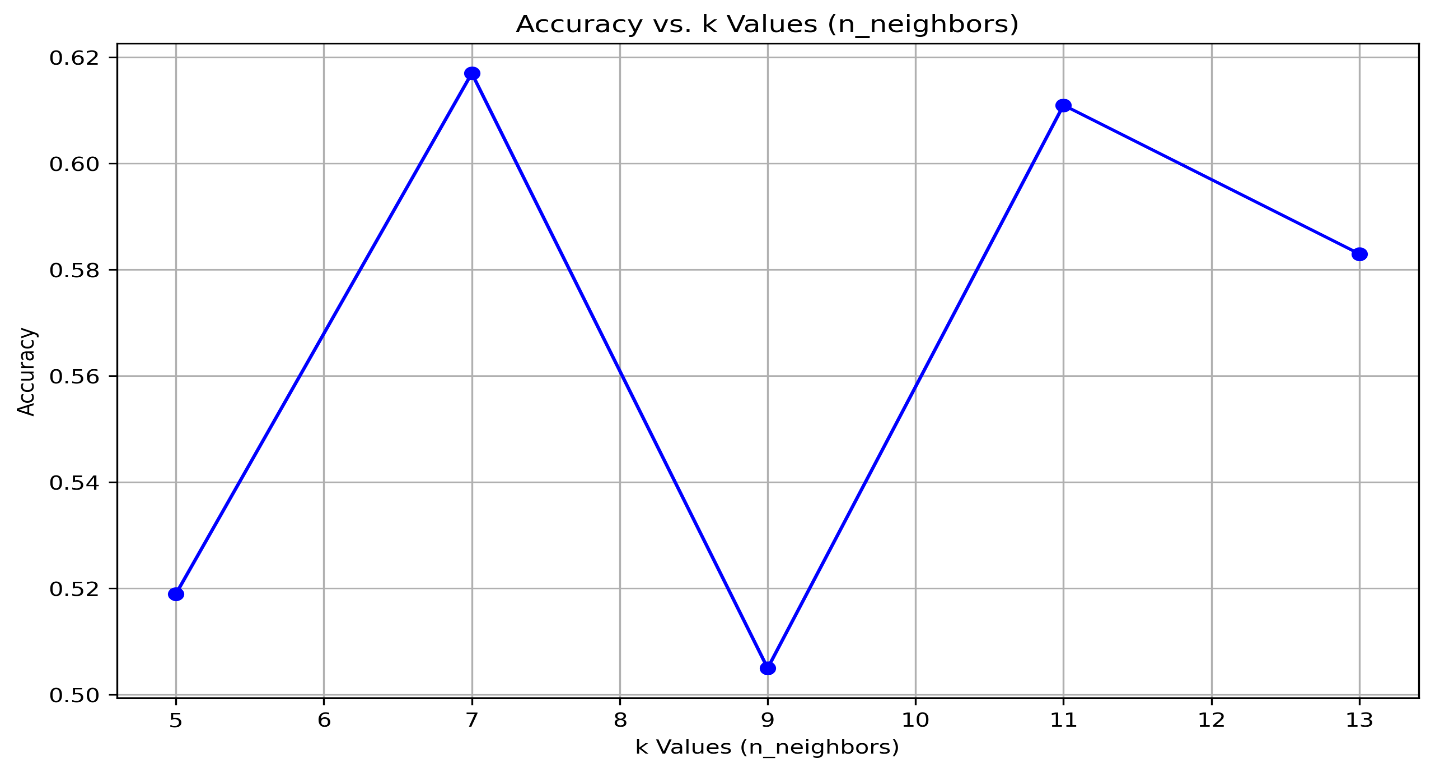
**Model Evaluation:**

* For each combination of hyperparameters and split, the KNN model was trained and tested.
* The model's performance was evaluated using key metrics, including accuracy, precision, and recall.

**Results:**

* The best combination of hyperparameters that achieved the highest accuracy for each training-test split was identified. Results were summarized in a tabulated format, showcasing the key hyperparameters and corresponding performance metrics for each split. Best k value is 7.





**Task 2.2:** **Study the impact of at least one key parameter of the model. Describe your findings. Choose the best value(s) for the parameter(s) and justify your choice.**

**Answer:** Based on the results provided, it appears that the key parameter under consideration is "n\_neighbors" in the k-nearest neighbors (KNN) algorithm. The best value for "n\_neighbors" seems to vary across the different train/test splits. To determine the overall best value, with the highest accuracy, which is in the 0.2 train/test split with "n\_neighbors" set to 7, resulting in an accuracy of 0.617. In the 0.2 train/test split, "n\_neighbors" = 7 achieved the highest accuracy, indicating that using 7 nearest neighbors for classification in this specific split resulted in the best performance.

A screenshot of a computer

Description automatically generated

### **Task 2.3: With the above optimal parameter(s), train and test the model on different training/test data splits: 20:80, 30:70, 40:60, 50:50, 60:40, 70:30, 80:20. What is the best train/test split? Why?**

## Answer: The best train/test split will be the one that results in the highest accuracy value, indicating the best model fit to the data is 80% training data and 20% test data.

A screenshot of a computer

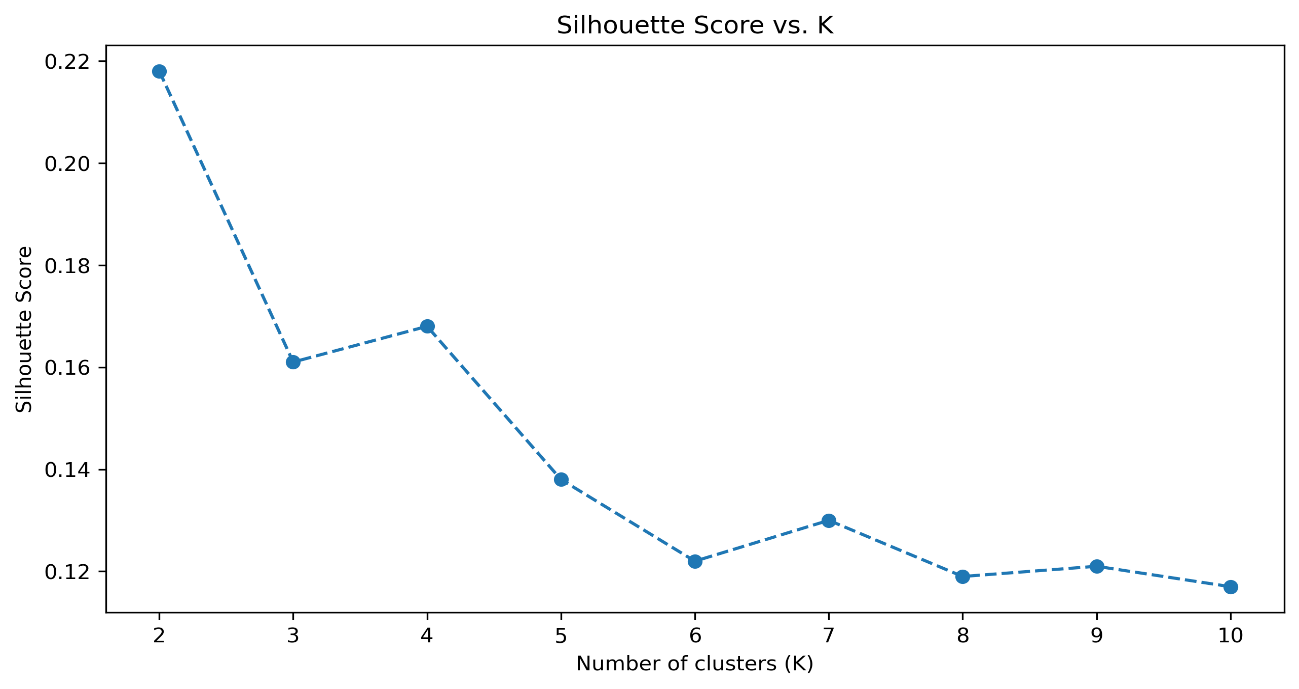
Description automatically generated

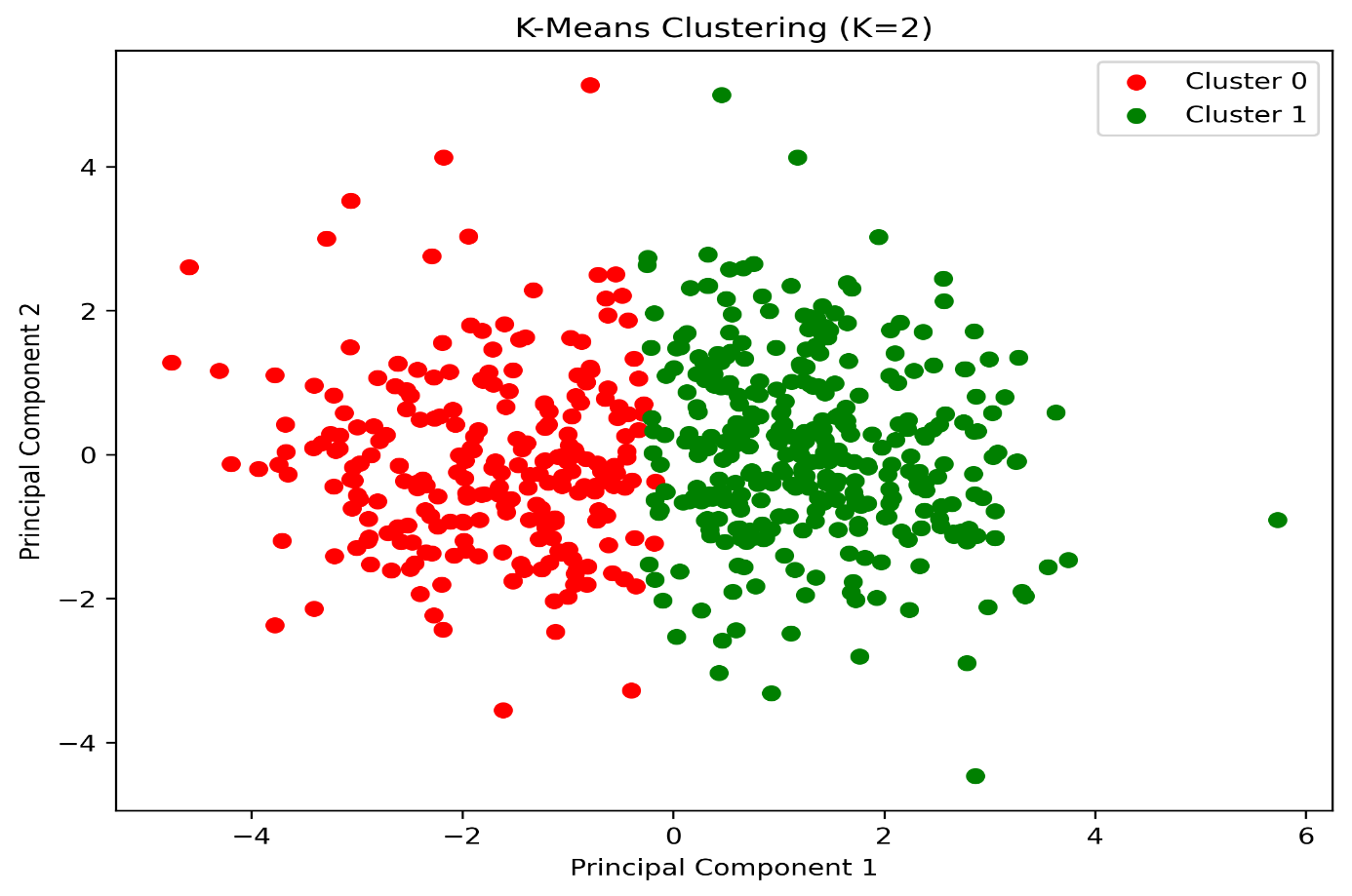
# Task 3: Classification

**Task 3.1. Select a model, either k-means or DBSCAN. Build and evaluate the model. Tune the key parameter(s) of the model and justify your choice of the value(s).**

**K-Means Clustering:**

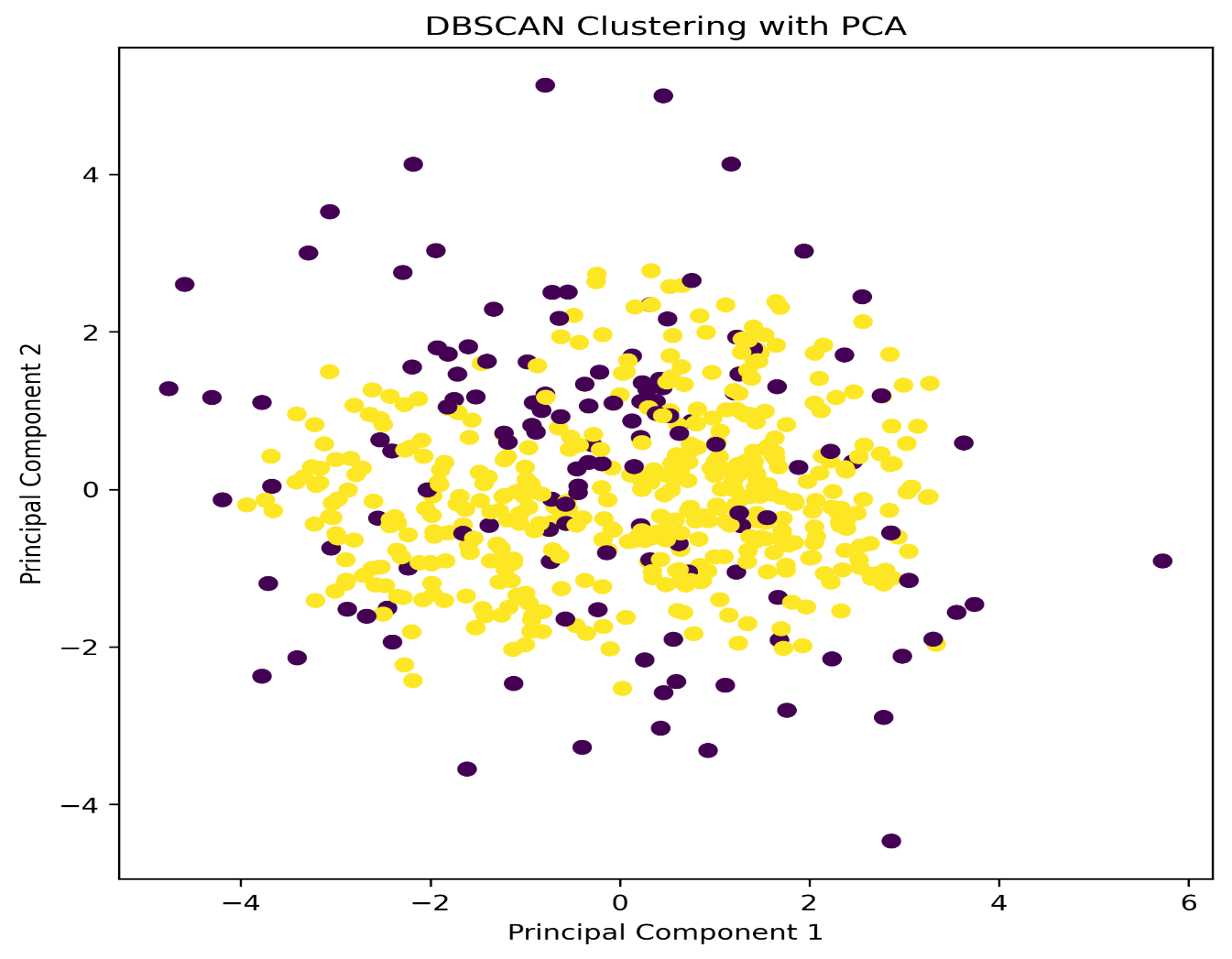
* The optimal number of clusters (K) was determined using the Silhouette Score, which measures the quality of clustering. The results were computed for K values ranging from 2 to 10.
* The table displays the Silhouette Scores for different K values, and it can be observed that the highest score is achieved when K is 2, indicating the best clustering result.
* A line graph was created to visualize the relationship between K values and Silhouette Scores. The graph shows that the Silhouette Score is highest for K=2.
* The K-Means clustering model was then trained with K=2, and the clusters were visualized using PCA.





**DBSCAN (Density-Based Spatial Clustering of Applications with Noise):**

* For DBSCAN, optimal hyperparameters (eps and min\_samples) were tuned to maximize the Silhouette Score.
* A combination of different eps values (ranging from 0.1 to 2.0) and min\_samples values (ranging from 5 to 19) was explored.
* The combinations that resulted in only one cluster were filtered out.
* The table displays the best combinations for each eps value, and the overall best combination was selected.
* DBSCAN was then applied with the best hyperparameters (best\_eps and best\_min\_samples), and the number of clusters was determined.
* The clusters were visualized using PCA.
* In summary, for K-Means Clustering, K=2 was found to be the optimal number of clusters based on the Silhouette Score. For DBSCAN, the best hyperparameters (eps and min\_samples) resulted in an optimal clustering configuration. The clusters for both K-Means and DBSCAN were visualized using PCA for further analysis and interpretation.



**Task 3.2: Determine the optimal number of clusters, and justify.**

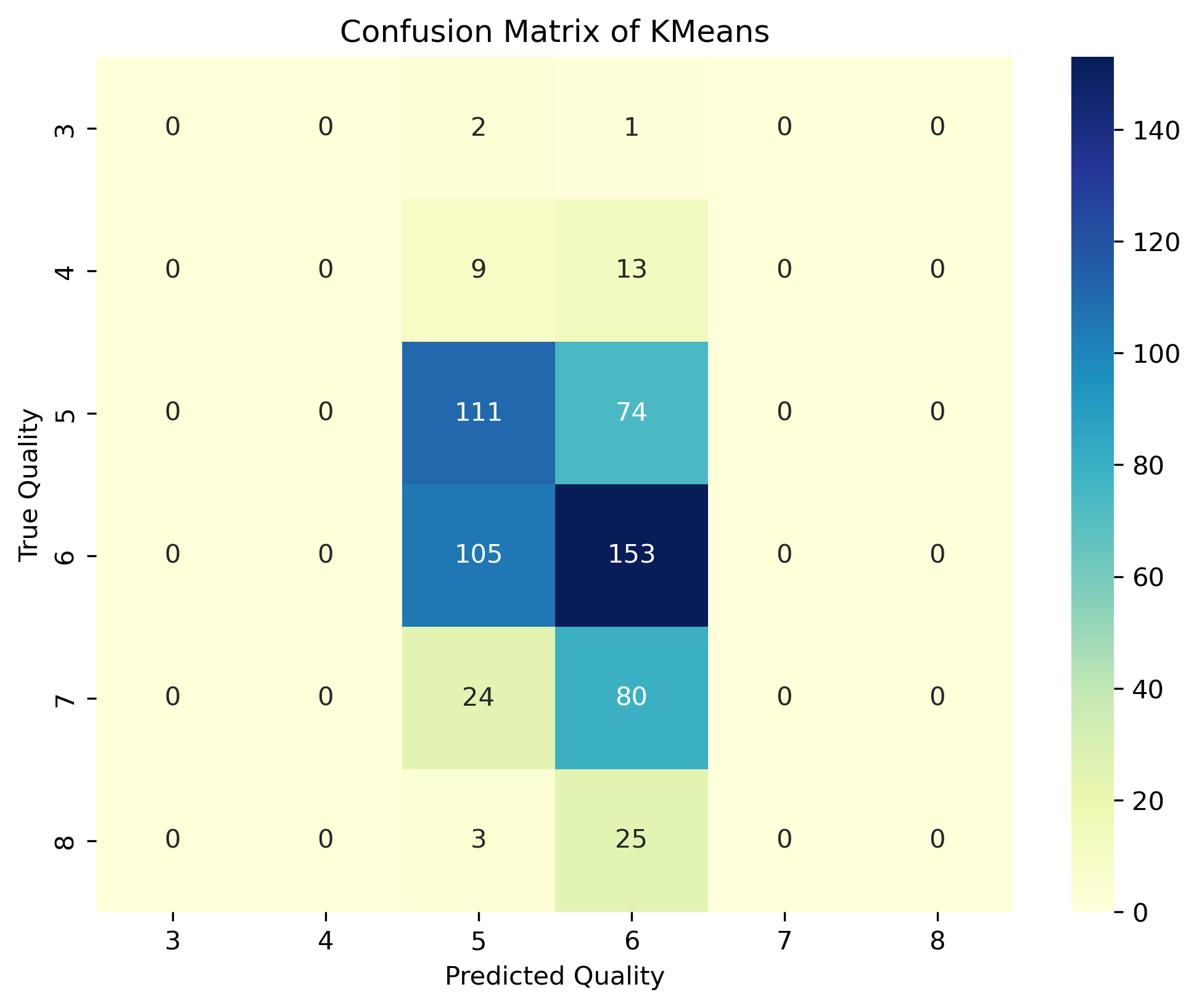
#### Answer: For KMeans Clustering, based on the Silhouette Score, K=2 is the optimal number of clusters as it has the highest score. For DBSCAN, based on the Silhouette Score, the optimal combination of hyperparameters for DBSCAN is achieved when Eps=1.9 and Min Samples=5, resulting in a higher Silhouette Score, indicating better cluster separation. This combination is the best choice for clustering, as it maximizes data point cohesion within clusters and separation between clusters.

**Task 3.3. Analyse the meaning (i.e., predicted quality level) of each cluster by checking the clustering results against the true quality levels (i.e., the variable quality). Construct and explain the confusion matrix of the results.**

**K-Means Clustering Confusion Matrix:**

In this case,

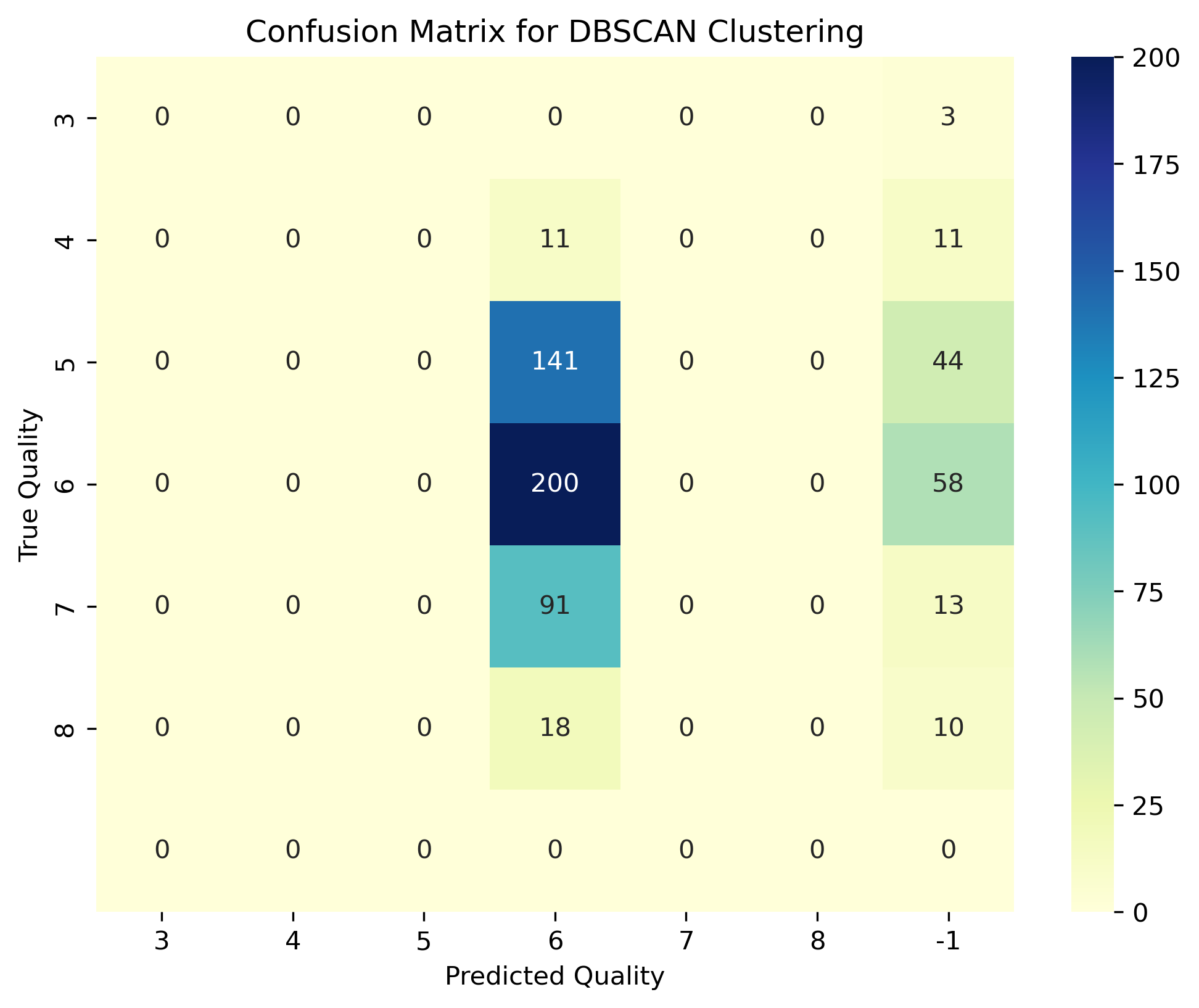
* Misclassifications: The confusion matrix indicates that there were several misclassifications, where data points from one quality level were assigned to another quality level by the K-Means clustering algorithm.
* Difficulty in Discrimination: It's apparent that some quality levels, such as 3 and 4, or 5 and 6, were frequently confused with each other, as shown by the higher counts in off-diagonal cells, indicating the model's difficulty in discriminating between these quality levels. different quality labels, making it challenging to assign a single quality label to each cluster.



**DBSCAN Confusion Matrix:**

Here's what the confusion matrix shows:

The diagonal elements (from top-left to bottom-right) represent the correctly classified samples for each quality level. For example, on the diagonal element at [3,3], there are 3 samples from the true quality level 3 that were correctly classified as 3.The off-diagonal elements represent the misclassified samples. For example, in the row for true quality 3, there are 3 samples that were misclassified as quality 7. In the row for true quality 4, 11 samples were misclassified as quality 6, and 11 samples were correctly classified as quality 4.The last column (with quality -1) represents the noise points. In the last row, there are 0 true quality points that were misclassified as noise, and 0 true noise points that were correctly identified as noise.



References

scikit-learn.org

https://matplotlib.org/stable/users/explain/colors/colormaps.html

<https://en.wikipedia.org/wiki/Silhouette_(clustering)#:~:text=The%20silhouette%20value%20is%20a,poorly%20matched%20to%20neighboring%20clusters>.

https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning